Comparative analysis of Deep Reinforcement Learning configurations in Flow Shop for enhanced Maintenance Management

Maria Grazia Marchesano*, Guido Guizzi*, Liberatina Carmela Santillo*

**Università degli Studi di Napoli "Federico II", Dipartimento di Ingegneria Chimica, dei Materiali e della Produzione Industriale, P.le Tecchio, 80, 80125- Napoli- Italy*

(mariagrazia.marchesano@unina.it, g.guizzi@unina.it, santillo@unina.it)

Abstract: Optimising maintenance scheduling in flow shop settings is a significant problem in achieving the goal of efficiency in industrial environments, necessitating novel solutions. This paper presents a complete multi-method approach to overcoming the complexities of maintenance management in flow shop production systems, combining Deep Reinforcement Learning (DRL) with advanced simulation approaches. We especially look into the effect of different configurations on the performance of a DRL-trained model tasked with maintenance decision-making. Our methodology comprises developing and comparing two distinct DRL configurations. We conducted rigorous simulation-based studies to assess the effectiveness of each DRL configuration in managing maintenance schedules under varied production needs and machine failure rates. The comparison research provides findings about the trade-offs between short-term efficiency and long-term sustainability in maintenance planning, emphasising sophisticated DRL techniques' ability to adaptively balance these goals. Our findings show that a multi-method approach combining DRL and simulation can provide a versatile and powerful tool for enhancing maintenance procedures in flow shop environments. By demonstrating the benefits and limitations of various DRL setups, the research adds vital perspectives to the ongoing development of intelligent production management systems, opening the path for more resilient and efficient manufacturing operations.

Keywords: Flow-shop, Deep Reinforcement Learning, Maintenance

1.Introduction

Maintenance planning within a production plant is a fundamental activity to ensure production efficiency and the achievement of quality and safety standards (Geurtsen et al., 2023) . In this context, we observe how complex and multi-critical the decision-making process is, considering all the factors that contribute to defining an optimal maintenance plan (Converso et al., 2023; Ogunfowora & Najjaran, 2023). Moreover, in the broader scope of maintenance, it is essential to address product management and end-of-life issues. This includes adopting sustainable practices and complying with current regulations (Gamberini *et al.*, 2008; Popolo *et al.*, 2022).

The factors that contribute to the decision complexity of the maintenance scheduling problem include balancing productivity goals and minimizing system downtime (Paz & Leigh, 1994). This issue is particularly critical in flow shops, where operations follow a specific production flow, and the interruption of one machine affects the entire line(Mao et al., 2021).

Traditionally, heuristic linear programming algorithms and genetic algorithms have been used to address this problem (Abate et al., 2023; Branda et al., 2021). However, traditional methods often have high computational times and do not allow for possible re-planning based on what happens in the production system (unforeseen breakdowns, micro-stops, etc.). To overcome the limitations of traditional approaches, the use of Deep Reinforcement Learning (DRL) is proposed (Nguyen et al., 2022). When tackling problems with DRL, the choice of modelling the characteristic elements is crucial: the learning agent and the environment. In particular, it is essential to accurately characterize how training takes place by defining the state space, the action space, and the reward function.

In this paper, we propose a comparative analysis of two different configurations of DRL elements for maintenance planning in a flow shop. Specifically, we will investigate which configuration best balances productivity and downtime parameters in a flow shop. To do so with combine DRL with simulation using a multi-method approach.

This analysis helps to determine which configurations are most appropriate and which characteristics of the system under consideration best describe it, guiding the learning agent to make the best choices in terms of maintenance planning.

The organization of this paper is as follows: Section 2 discusses the literature review. Section 3 describes the problem statement and the tool used in this research, the

Section 4 focuses on the proposed approach and system settings and finally, Section 5 covers the experimental results and the discussion. The paper concludes with Section 6, where there is the summarization of the main findings.

2.Literature review

The integration of machine learning techniques with traditional optimization strategies has significantly advanced the field of maintenance scheduling. These developments aim to enhance operational efficiency and reduce production disruptions. Among the various machine learning approaches, deep reinforcement learning (DRL) techniques, such as Deep Q-Network (DQN) and Proximal Policy Optimization (PPO), have emerged as particularly promising.

Valet et al. (2022) (Valet et al., 2022) demonstrated the use of DQN for opportunistic maintenance scheduling, where maintenance tasks are planned during low-demand periods to minimize downtime. This study effectively highlighted DQN's ability to optimize sequential decisions in uncertain environments. However, it lacked real-time adaptability to unexpected machine breakdowns, an aspect our research addresses through dynamic re-planning capabilities.

In the realm of digital twin-enabled manufacturing systems, Yan, et al. (2022) (Yan et al., 2022) introduced a double-layer Q-learning algorithm for dynamic scheduling. While innovative, their approach did not compare their approach with other DRL algorithms limiting its comparative value. Our study bridges this gap by directly comparing DQN and PPO, providing a clearer understanding of their relative advantages in maintenance scheduling.

Mao et al. (2022) (J. Y. Mao et al., 2022) explored a hash map-based memetic algorithm for scheduling, suggesting the potential benefits of integrating DQN to enhance decision-making with real-time data. Although the study did not implement DRL, it underscored the need for dynamic system responses, which our research incorporates through direct application of DRL techniques.

Huang et al. (2020) (Huang et al., 2020) proposed a preventive maintenance policy using DQN for serial production lines, demonstrating DQN's effectiveness in balancing maintenance costs and machine availability. While their focus was on serial lines, our research extends the application to flow shops, thereby broadening the understanding of DRL in different production settings.

Akl et al. (2022) (Akl et al., 2022) developed a simulation-optimization approach that, although not utilizing DRL, emphasized the integration of strategic planning activities. This could benefit from the capabilities of PPO, which we explore in our research to enhance decision-making in maintenance scheduling.

Addressing broader challenges in predictive maintenance, Nunes et al. (2023) (Nunes et al., 2023) reviewed the challenges in predictive maintenance, emphasizing the need for precise data and effective machine learning models. They highlighted the potential of DRL, particularly PPO, in improving predictive maintenance strategies. Our study builds on these insights by empirically testing PPO's effectiveness in a flow shop maintenance context.

Kosanoglu et al. (2022) (Kosanoglu et al., 2022) presented a DRL-assisted simulated annealing algorithm, blending the strengths of simulated annealing with DRL's robust optimization capabilities, particularly PPO. However, they did not compare this with other DRL algorithms like DQN. Our research fills this gap by providing a direct comparison between DQN and PPO.

Nguyen et al. (2022) (Nguyen et al., 2022) utilize a multi-agent DRL framework that employs PPO to enhance decision-making across large-scale systems, illustrating PPO's effectiveness in managing multiple agents and complex operational demands, leading to improved maintenance decision-making and system reliability.

Hu et al. (2023) (Hu et al., 2023) explore the integration of PPO in their Knowledge Enhanced Reinforcement Learning (KERL) approach for optimizing production and maintenance scheduling. Their work demonstrates PPO's superior ability to manage multiple constraints and uncertainties, yielding higher business rewards and more effective failure prevention.

Ruiz Rodríguez et al. (2022) and Ruiz-Rodríguez et al. (2024) (Ruiz Rodríguez et al., 2022; Ruiz-Rodríguez et al., 2024) also reflect on the use of multi-agent reinforcement learning, with a strong indication of employing techniques like PPO to coordinate maintenance scheduling under uncertainty, achieving significant improvements in downtime and failure prevention.

Lastly, Xu et al. (2024) (Xu et al., 2024)detail a condition-based maintenance model using a factored Markov decision process, where an online reinforcement learning algorithm, potentially integrating DQN, efficiently learns and optimizes maintenance actions, showcasing the method's ability to adapt to and predict system needs with high accuracy.

This extensive body of research underscores the transformative potential of advanced AI techniques such as DQN and PPO in refining maintenance strategies within manufacturing. However, gaps remain in comparative analyses and real-time adaptability. Our study addresses these gaps by directly comparing DQN and PPO in a flow shop context, emphasizing their practical implications for dynamic maintenance scheduling.

3. Problem Statement

In modern manufacturing systems, maintenance planning is crucial to ensure high productivity and minimal downtime. There are two main types of maintenance activities:

- Preventive Maintenance: Scheduled activities aimed at preventing equipment failures.

- Corrective Maintenance: Reactive activities performed after a breakdown has occurred.

Balancing preventive maintenance with production planning is a complex task. The goal is to schedule maintenance activities in a way that minimizes disruptions to production while preventing unexpected equipment failures. This problem becomes more complex when considering:

- The stochastic nature of machine breakdowns.
- The varying impact of maintenance activities on production schedules.
- The need to balance short-term production goals with long-term equipment reliability.

This complexity can be effectively modelled using Reinforcement Learning (RL) techniques, where the problem is framed as a Markov Decision Process (MDP). In an MDP, the environment is represented by states, actions, and rewards, which align well with the decision-making process in maintenance scheduling.

3.1 Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL)

In this sub-section, we present the RL and DRL methodologies used to address the problem and detail the two algorithms compared in this study: DQN and PPO.

Reinforcement Learning (RL) is a type of machine learning where an agent learns to make decisions by performing actions within an environment to maximize cumulative reward. The agent interacts with the environment in discrete time steps, observes the state of the environment, selects, and performs actions, receives rewards, and updates its knowledge based on these interactions (Figure 1).

Deep Reinforcement Learning (DRL) combines RL with deep learning techniques to oversee high-dimensional state and action spaces. DRL leverages deep neural networks to approximate value functions or policies, enabling the agent to make more complex decisions. This approach is particularly useful in environments where the state space is too large for traditional RL methods.

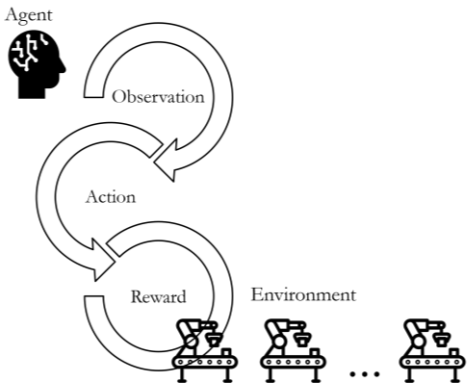

Figure 1 Reinforcement Learning scheme.

In this paper, we aim to compare the two most widely used algorithms in the field of DRL: Deep Q-Network and Proximal Policy Optimization. We will examine how these algorithms are characterised and how they apply to the case study presented in this article. Through this comparison, we seek to highlight the situations in which one algorithm outperforms the other in terms of minimising downtime and maximising throughput.

3.2 Deep Q-Network (DQN)

Deep Q-Network (DQN) is a reinforcement learning algorithm that integrates Q-learning with deep neural networks to manage high-dimensional state space. Developed by Mnih et al., 2015, DQN uses a neural network to approximate the Q-value function. The Q-value function quantifies the quality of a state-action pair, providing an estimate of the total reward that can be obtained by starting at a given state, taking a particular action, and following a certain policy thereafter.

DQN employs a deep neural network to learn the representation of the state, providing a more generalizable approach to Q-learning.

To break the correlation between consecutive samples and to use the learning data more efficiently, DQN utilizes an experience replay mechanism. Actions and states are stored in a replay buffer and sampled randomly to train the network.

To stabilize the learning process, DQN uses a separate target network with the same architecture as the primary network but with frozen parameters. These parameters are updated less frequently to prevent the moving target problem, where updates lead to significant changes in policy.

During training, the agent interacts with the environment, and the transition tuples (s,a,r,s') are stored in the replay buffer. The network is trained by minimizing the loss between predicted Q-values and the target Q-values, which are computed using the reward and the maximum predicted Q-value of the next state, discounted by a factor γ (gamma).

3.3 Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO) is another influential algorithm in the field of reinforcement learning, developed for solving problems with large and continuous action spaces (Schulman et al., 2017). It belongs to the family of policy gradient methods where the objective is to directly optimize the policy function. PPO aims to address the key challenges of policy gradient methods, such as sample inefficiency and high sensitivity to hyperparameters, by employing a novel objective function that facilitates stable and efficient learning.

PPO introduces a clipping mechanism in the objective function to prevent excessively large policy updates. This is achieved by modifying the typical policy gradient objective to include a term that minimizes the deviation between new and old policies.

Unlike standard policy gradient methods, PPO exploits the collected data for several epochs of stochastic gradient ascent, enhancing data efficiency.

Another variant of PPO uses a penalty to regulate the updates, ensuring that the new policy is not too far from the old. PPO operates in two phases, in the first phase, data is collected through interaction with the environment using the current policy. In the second phase, the policy is updated by optimizing the clipped objective function. This process iteratively refines the policy towards optimal behaviour.

3.4 Challenges of Planning Maintenance Activities in a Flow Shop

Planning maintenance activities in a flow shop environment presents several unique challenges:

- High Interdependence of Operations: In a flow shop, the production process is sequential, meaning a failure in one machine can cause a cascade of delays throughout the entire line. This interdependence requires maintenance schedules to be highly coordinated to minimize overall downtime.
- Balancing Throughput and Maintenance: Maintenance activities are essential to prevent unexpected breakdowns, but they also take machines out of operation, directly impacting production throughput. Finding the optimal balance between these competing priorities is critical.
- Variable Production Demands: Flow shops often face fluctuating production demands, requiring dynamic maintenance scheduling that can adapt to varying workloads and priorities without compromising equipment reliability.
- Resource Constraints: Maintenance resources, including personnel and spare parts, are often limited. Effective maintenance planning must optimize the use of these resources to ensure maximum availability of machinery.
- Impact on Makespan: The total time required to complete a set of jobs, known as makespan, is a critical metric in flow shop scheduling. Maintenance activities must be scheduled in a way that minimizes their impact on the makespan, ensuring that production deadlines are met and overall efficiency is maintained.

By addressing these challenges through the application of DQN and PPO within our proposed DRL framework, we aim to develop a maintenance scheduling system that is both efficient and adaptable to the complex dynamics of flow shop environments. This approach seeks to minimize downtime and makespan, thereby enhancing overall productivity and operational resilience.

4. Proposed Approach

Building on the comparative analysis of Deep Q-Network and Proximal Policy Optimization, our proposed approach integrates DRL with simulation methods to create a comprehensive framework for optimizing maintenance decisions. These algorithms are chosen for their ability to manage high-dimensional state spaces and large, continuous action spaces, respectively. By leveraging DQN's experience replay mechanism and target network stabilization, alongside PPO's novel objective function and clipping mechanism, we aim to enhance the robustness and efficiency of maintenance scheduling. Rigorous simulation-based studies are employed to assess the performance of each configuration under varying production demands and machine failure rates, providing valuable insights into the trade-offs between short-term efficiency and long-term sustainability. This multi-method approach underscores the potential of DRL techniques to dynamically balance maintenance objectives, contributing to more resilient and efficient manufacturing operations.

4.1 System configuration

The proposed methodology incorporates the use of AnyLogic simulation software to model a production line, which enables the simulation of line operations, corrective, and preventive maintenance events. This setup is critical for comparing the efficacy of DQN and PPO algorithms in training a reinforcement learning agent. The AnyLogic software enables the configuration of key parameters necessary for a Reinforcement Learning Experiment. Once configured, the model will be exported and implemented in Python using the ALPyne library.

During the simulation, the DQN and PPO algorithms will be developed in Python to train an agent on a flow shop system aimed at deriving an optimal policy. The AnyLogic environment is specifically tailored to address the Flow Shop Scheduling Problem (FSSP), enabling experimentation with various configurations (i.e. number of jobs to be processed, number of machines in the line). The hypothesis of the simulation model are the one taken from the work Branda et al., 2021, in which the authors wish to schedule at the same time production and maintenance activities using Meta-Heuristic algorithms.

The simulation model's parameters are the following:

- Number of jobs to be processed: 50.
- Processing Times: Triangular distribution with a minimum of 20, maximum of 100, and mode of 50.
- Corrective Maintenance Time: Uniform distribution (15,25) minutes.
- Preventive Maintenance Time: Uniform distribution (30,50) minutes.
- Weibull scale parameters $\alpha = 100$ and $\beta = 1.2$.

ALPyne, a Python library, facilitates the interactive execution of RL models exported from AnyLogic. Since AnyLogic lacks native support for RL algorithms, ALPyne

is crucial for running these experiments. Configuration within ALPyne includes setting up the observation and action spaces and managing the conversion of Observation objects and actions based on the data types used in the AnyLogic model. Specifically, the observation space is modelled as a 5-dimensional continuous space. The action space is discrete, representing maintenance decisions (0 for no maintenance, 1 for maintenance). The reward function is constructed to evaluate the agent's performance by balancing the Makespan against the frequency of maintenance tasks. This involves optimizing individual ratios related to maintenance times and failure probabilities, crucial for maximizing product performance and minimizing operational downtime.

The state space S includes key parameters that influence the maintenance scheduling decisions:

- Failure Probabilities: Machine failure probabilities.
- Percentage of Completed Jobs: Jobs completed at time t.
- Corrective Maintenance Ratio: The ratio of the mean processing time to the sum of the processing time and the corrective maintenance time at time t.
- Preventive Maintenance Ratio: The ratio of the sum of the processing time and the time in preventive maintenance to the mean processing time at time t.
- Total Maintenance Ratio: The ratio of the total processing time at time t to the sum of the total processing time, corrective time, and preventive time at time t.

The action space A is binary, representing whether to perform preventive maintenance (1) or not (0) on each machine.

The goal is to minimize the makespan while limiting the number of machine breakdowns. This requires balancing preventive maintenance interventions to avoid excessive downtime.

The reward function is defined to calculate the reward for the agent based on the observation received from the simulation as: Reward = Failure Probabilities · Corrective Maintenance Ratio · Preventive Maintenance Ratio · Total Maintenance Ratio.

To maximize performance, the learning agent aims to optimize the individual ratios. Maximizing Corrective Maintenance Ratio involves minimizing the number of failures, while maximizing Preventive Maintenance Ratio requires increasing the number of maintenance tasks. This is balanced by the optimization of Total Maintenance Ratio. The overall reward is influenced by the failure probability, guiding the agent towards optimal maintenance strategies.

The environment is stochastic, with processing times and maintenance times (both preventive and corrective) being stochastic variables. This variability reflects real-world uncertainties in the manufacturing process. The simulation model, developed using AnyLogic and executed through the ALPyne library in Python, incorporates these dynamics to create a realistic and challenging training environment for the RL agent. The parameters and settings allow for comprehensive experimentation and evaluation of the DQN and PPO algorithms in optimizing maintenance scheduling within a flow shop system.

5.Results and Discussion

After the development of the two policies (DQN and PPO) they were assessed in the simulation environment in order to compare the results obtained in terms of makespan, maintenance frequency and the number of breakdowns. These results were compared to a heuristic approach from the literature. The setting of the experiment of the latter cited approach (Branda et al., 2021a problem number "201") was replicated in the Anylogic simulation model.

Table 1 Simulation results

Model	Makespan	Maintenance frequency	Number of Breakdown
DQN	3239.0	6.0	3.0
PPO	3345.0	9.0	5.0
Heuristic (GA)	2834.0	1.0	11.0

Both the DQN and PPO models have relatively high makespan with the Heuristic model designed to operate more quickly, presumably through optimizations or shortcuts that reduce operational time. The DQN and PPO models likely have differing maintenance frequencies due to their handling of risks, with PPO potentially requiring more frequent checks or interventions due to its higher failure probabilities. The Heuristic model, while having a lower frequency of maintenance, pays for this with a higher rate of breakdowns.

Number of Breakdown is a critical metric for operational reliability. The DQN model seems to balance efficiency and risk well, leading to fewer breakdowns. The PPO model, with its deeper exploration strategies, shows a moderate number, and the Heuristic model experiences the most, as its lower maintenance frequency and faster operational tempo might lead to increased wear and tear or oversight of potential issues.

The analysis of the results comparing the two distinct reinforcement learning models, specifically DQN and PPO, reveals noteworthy distinctions in their operation and outputs. The makespan, which is the total time required to complete a set of jobs, is noted as 3239.0 time units. In comparison, the PPO records a makespan of 3345.0 time units. This indicates a slightly longer duration under the PPO model, suggesting a higher number of maintenance task to be chosen. The DQN model appears to balance risk and operational time more conservatively,

whereas the PPO model, while potentially achieving better long-term learning through exploration, might incur higher immediate costs in terms of time and risk.

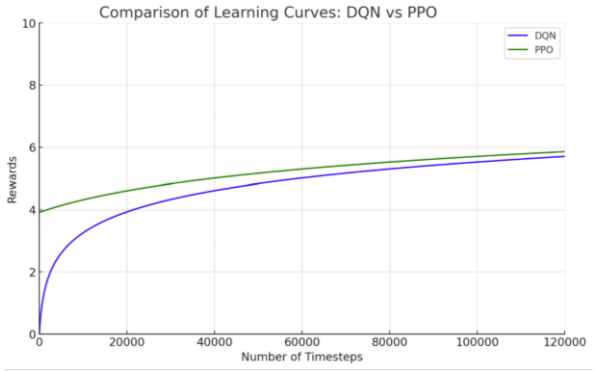

Figure 2 Comparison of the Learning Curves: DQN vs PPO

To effectively compare the two figures showcasing the learning curves for PPO and DQN training, we need to examine several aspects, including the rapidity of learning, the stability of rewards, and the overall performance achieved by each algorithm. The learning curve for PPO demonstrates a very rapid increase in rewards from the outset, quickly escalating from around 4 to just over 8 within the first 20,000 timesteps. This rapid ascent indicates that the PPO algorithm quickly grasps and adapts to the environment, optimizing its policy at an impressive rate. The DQN training curve shows a similar swift increase in reward values, rising from about 4 to surpass 8 within the same initial timeframe.

This parallel suggests that DQN, much like PPO, exhibits strong initial learning capabilities, efficiently capturing optimal strategies early in the training process. Beyond the initial phase, the PPO curve displays a stable reward pattern, with values slightly fluctuating around the 8 to 9 range throughout the remaining training period. This stability implies that once PPO attains a high-performance level, it consistently maintains this level, with minor fluctuations reflecting its ongoing exploration of the policy space. The DQN curve also stabilizes after the initial learning phase but displays lower values.

Both algorithms demonstrate good capabilities in terms of rapid learning and achieving high rewards. The curves validate the effectiveness of employing advanced reinforcement learning techniques to optimize complex decision-making processes in dynamic environments.

6.Conclusions

The present research adopts an innovative strategy to tackle the complexities of scheduling maintenance tasks in a Flow Shop environment, where traditional scheduling tools often fall short. This gap has paved the way to the exploration of new methodologies, such as deep reinforcement learning (DRL), which offers promising solutions for optimizing maintenance planning in manufacturing.

This study specifically examines the performance of two DRL algorithms, Deep Q-Network (DQN) and Proximal Policy Optimization (PPO), in their ability to efficiently schedule maintenance tasks. The findings highlight a significant reduction in the number of maintenance operations required compared to traditional methods, demonstrating the potential and effectiveness of the proposed DRL-based approach.

Future research will focus on a broader experimental plan to deepen our understanding of DRL in maintenance planning across more complex multi-machine, multi-product production systems. This will include investigating distinct levels of stochasticity within the model and integrating considerations for resource allocation—such as manpower, equipment, and other essential resources—into the maintenance planning process. By addressing these resource constraints, the study aims to develop a more realistic and applicable strategy for maintenance scheduling.

References

Abate, R., Vespoli, S., Guizzi, G., & Marchesano, M. G. (2023). Dynamic Optimization of Pickup and Delivery Problems in Industry 4.0: A Vickrey Auction-Based and Multi-Agent Simulation Approach. Proceedings of the Summer School Francesco Turco.

Akl, A. M. *et al.* (2022) 'A Joint Optimization of Strategic Workforce Planning and Preventive Maintenance Scheduling: A Simulation–Optimization Approach', *Reliability Engineering and System Safety*, 219(November 2021), p. 108175.

Branda, A. *et al.* (2021a) 'Dataset of metaheuristics for the flow shop scheduling problem with maintenance activities integrated', *Data in Brief*, 36, p. 106985.

Branda, A. *et al.* (2021) 'Metaheuristics for the flow shop scheduling problem with maintenance activities integrated', *Computers and Industrial Engineering*, 151(November 2020), p. 106989.

Converso, G., Gallo, M., Murino, T., & Vespoli, S. (2023). Predicting Failure Probability in Industry 4.0 Production Systems: A Workload-Based Prognostic Model for Maintenance Planning. Applied Sciences (Switzerland), 13(3).

Gamberini, R., Gebennini, E., Grassi, A., Mora, C., & Rimini, B. (2008). An innovative model for WEEE recovery network management in accordance with the EU directives. International Journal of Environmental Technology and Management, 8(4), 348–368.

Geurtsen, M., Adan, I. and Atan, Z. (2023) 'Deep reinforcement learning for optimal planning of assembly line maintenance', *Journal of Manufacturing Systems*, 69(February), pp. 170–188.

Hu, J. *et al.* (2023) 'Knowledge-enhanced reinforcement learning for multi-machine integrated production and maintenance scheduling', *Computers & Industrial Engineering*, 185(March), p. 109631.

Huang, J., Chang, Q. and Arinez, J. (2020) 'Deep reinforcement learning based preventive maintenance policy for serial production lines', *Expert Systems with*

Applications, 160, p. 113701.

Kosanoglu, F., Atmis, M. and Turan, H. H. (2022) 'A deep reinforcement learning assisted simulated annealing algorithm for a maintenance planning problem', *Annals of Operations Research*.

Mao, J. Y. *et al.* (2022) 'A hash map-based memetic algorithm for the distributed permutation flowshop scheduling problem with preventive maintenance to minimize total flowtime', *Knowledge-Based Systems*, 242, p. 108413.

Mao, J. yang *et al.* (2021) 'An effective multi-start iterated greedy algorithm to minimize makespan for the distributed permutation flowshop scheduling problem with preventive maintenance', *Expert Systems with Applications*, 169(December 2020), p. 114495.

Mnih, V. *et al.* (2015) 'Human-level control through deep reinforcement learning', *Nature*, 518(7540), pp. 529–533.

Nguyen, V. T. *et al.* (2022) 'Artificial-intelligence-based maintenance decision-making and optimization for multi-state component systems', *Reliability Engineering and System Safety*, 228(July), p. 108757.

Nunes, P., Santos, J. and Rocha, E. (2023) 'Challenges in predictive maintenance – A review', *CIRP Journal of Manufacturing Science and Technology*, 40, pp. 53–67.

Ogunfowora, O. and Najjaran, H. (2023) 'Reinforcement and deep reinforcement learning-based solutions for machine maintenance planning , scheduling policies , and optimization', *Journal of Manufacturing Systems*, 70(July), pp. 244–263.

Paz, N. M. and Leigh, W. (1994) 'Maintenance Scheduling: Issues, Results and Research Needs', *International Journal of Operations & Production Management*, 14(8), pp. 47–69.

Popolo, V., Vespoli, S., Gallo, M., & Grassi, A. (2022). A systemic analysis of the impacts of Product 4.0 on the triple bottom-line of Sustainability. IFAC-PapersOnLine, 55(10), 1110–1115.

Ruiz-Rodríguez, M. L. *et al.* (2024) 'Dynamic maintenance scheduling approach under uncertainty: Comparison between reinforcement learning, genetic algorithm simheuristic, dispatching rules', *Expert Systems with Applications*, 248(October 2023), p. 123404.

Ruiz Rodríguez, M. L. *et al.* (2022) 'Multi-agent deep reinforcement learning based Predictive Maintenance on parallel machines', *Robotics and Computer-Integrated Manufacturing*, 78(July), p. 102406.

Schulman, J. *et al.* (2017) 'Proximal Policy Optimization Algorithms', pp. 1–12.

Valet, A. *et al.* (2022) 'Opportunistic maintenance scheduling with deep reinforcement learning', *Journal of Manufacturing Systems*, 64(March), pp. 518–534.

Xu, J. *et al.* (2024) 'Online reinforcement learning for condition-based group maintenance using factored Markov decision processes', *European Journal of Operational Research*, 315(1), pp. 176–190.

Yan, Q., Wang, H. and Wu, F. (2022) 'Digital twin-enabled dynamic scheduling with preventive maintenance using a double-layer Q-learning algorithm', *Computers and Operations Research*, 144(July 2021), p. 105823.